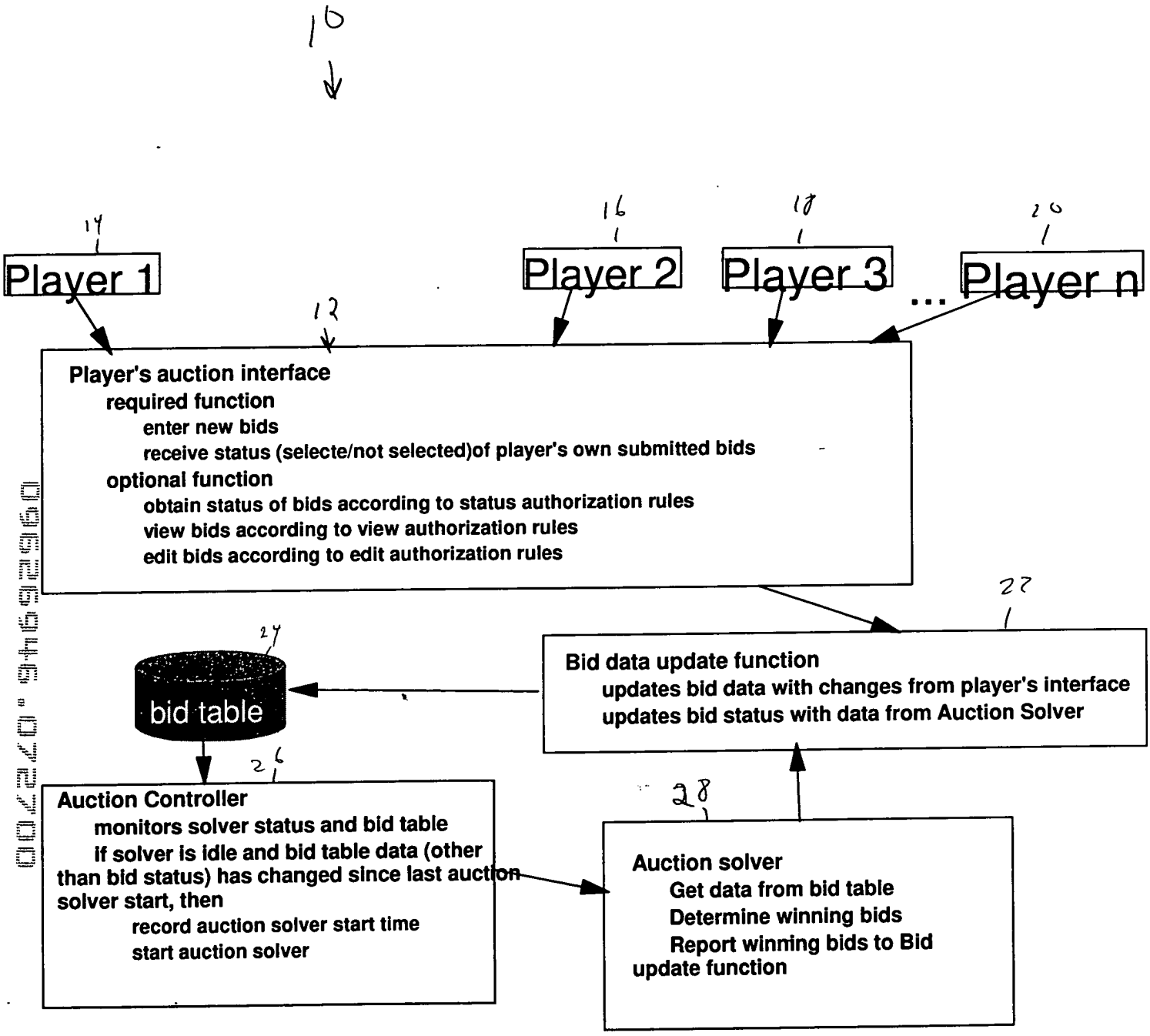
10
14
Player 1
16
Player 2
18
Player 3
...
20
Player n
12
Player's auction interface
required function
enter new bids
receive status (selecte/not selected)of player's own submitted bids
optional function
obtain status of bids according to status authorization rules
view bids according to view authorization rules
edit bids according to edit authorization rules
22
Bid data update function
updates bid data with changes from player's interface
updates bid status with data from Auction Solver
24
bid table
26
Auction Controller
monitors solver status and bid table
if solver is idle and bid table data (other than bid status) has changed since last auction solver start, then
record auction solver start time
start auction solver
28
Auction solver
Get data from bid table
Determine winning bids
Report winning bids to Bid update function

FIG 1

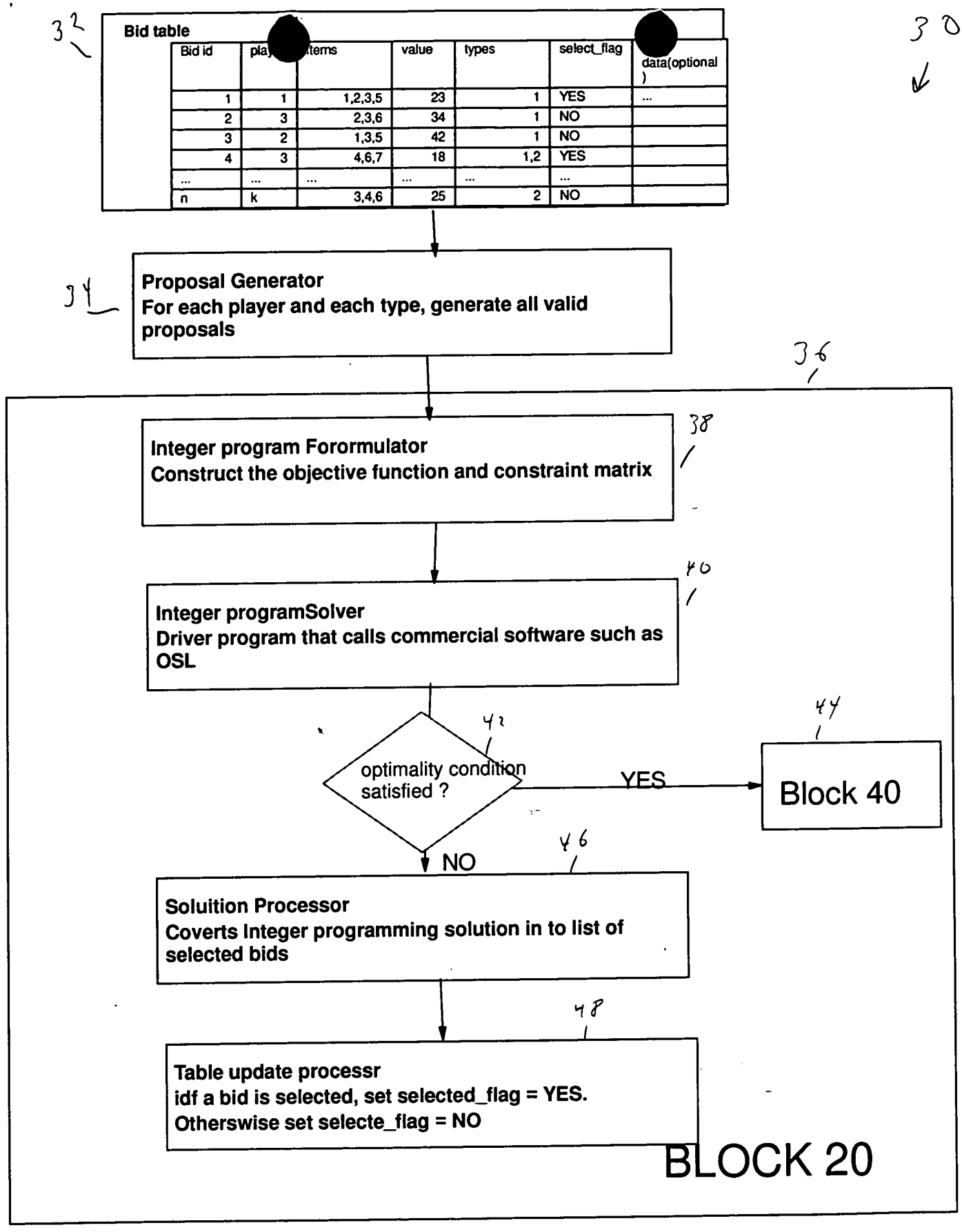
32
Bid table
Bid id
pla
tems
value
types
select_flag
data(optional)
1 1 1,2,3,5 23 1 YES ...
2 3 2,3,6 34 1 NO
3 2 1,3,5 42 1 NO
4 3 4,6,7 18 1,2 YES
... ...
n k 3,4,6 25 2 NO
30
34
Proposal Generator
For each player and each type, generate all valid proposals
36
38
Integer program Forormulator
Construct the objective function and constraint matrix
40
Integer programSolver
Driver program that calls commercial software such as OSL
42
optimality condition satisfied ?
YES
44
Block 40
NO
46
Soluition Processor
Coverts Integer programming solution in to list of selected bids
48
Table update processr
idf a bid is selected, set selected_flag = YES.
Otherswise set selecte_flag = NO
BLOCK 20

FIG 2

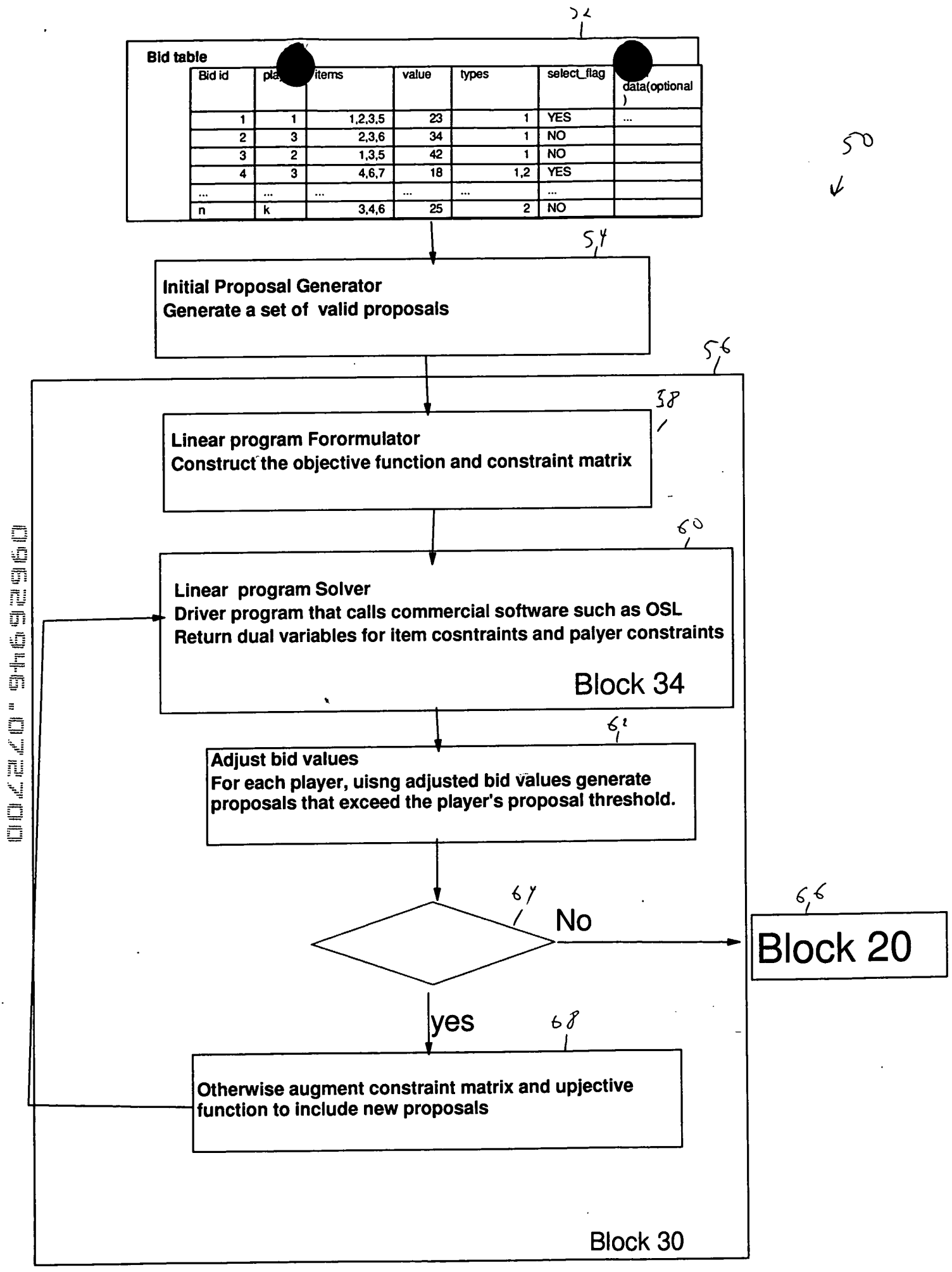
Bid table
Bid id
items
value
types
select_flag
data(optional)
1 1 1,2,3,5 23 1 YES ...
2 3 2,3,6 34 1 NO
3 2 1,3,5 42 1 NO
4 3 4,6,7 18 1,2 YES
... ...
n k 3,4,6 25 2 NO
50
52
54
Initial Proposal Generator
Generate a set of valid proposals
56
58
Linear program Fororformulator
Construct the objective function and constraint matrix
60
Linear program Solver
Driver program that calls commercial software such as OSL
Return dual variables for item cosntraints and palyer constraints
Block 34
62
Adjust bid values
For each player, uisng adjusted bid values generate proposals that exceed the player's proposal threshold.
64
No
66
Block 20
yes
68
Otherwise augment constraint matrix and upjective function to include new proposals
Block 30

FIG 3

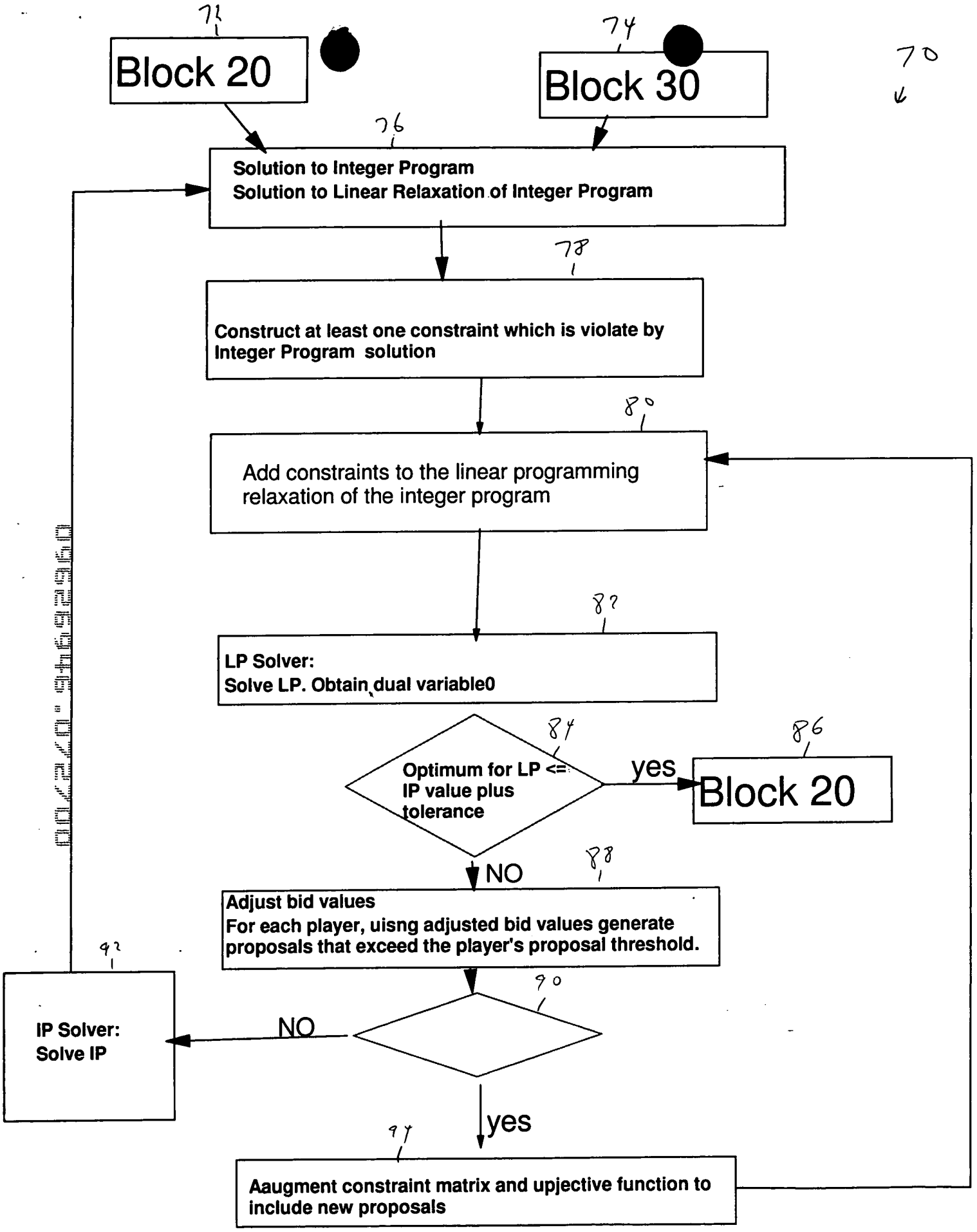
72
Block 20
74
Block 30
70
76
Solution to Integer Program
Solution to Linear Relaxation of Integer Program
78
Construct at least one constraint which is violate by Integer Program solution
80
Add constraints to the linear programming relaxation of the integer program
82
LP Solver:
Solve LP. Obtain dual variable0
84
Optimum for LP <= IP value plus tolerance
yes
86
Block 20
NO
88
Adjust bid values
For each player, uisng adjusted bid values generate proposals that exceed the player's proposal threshold.
90
NO
92
IP Solver:
Solve IP
yes
94
Aaugment constraint matrix and upjective function to include new proposals

FIG 4

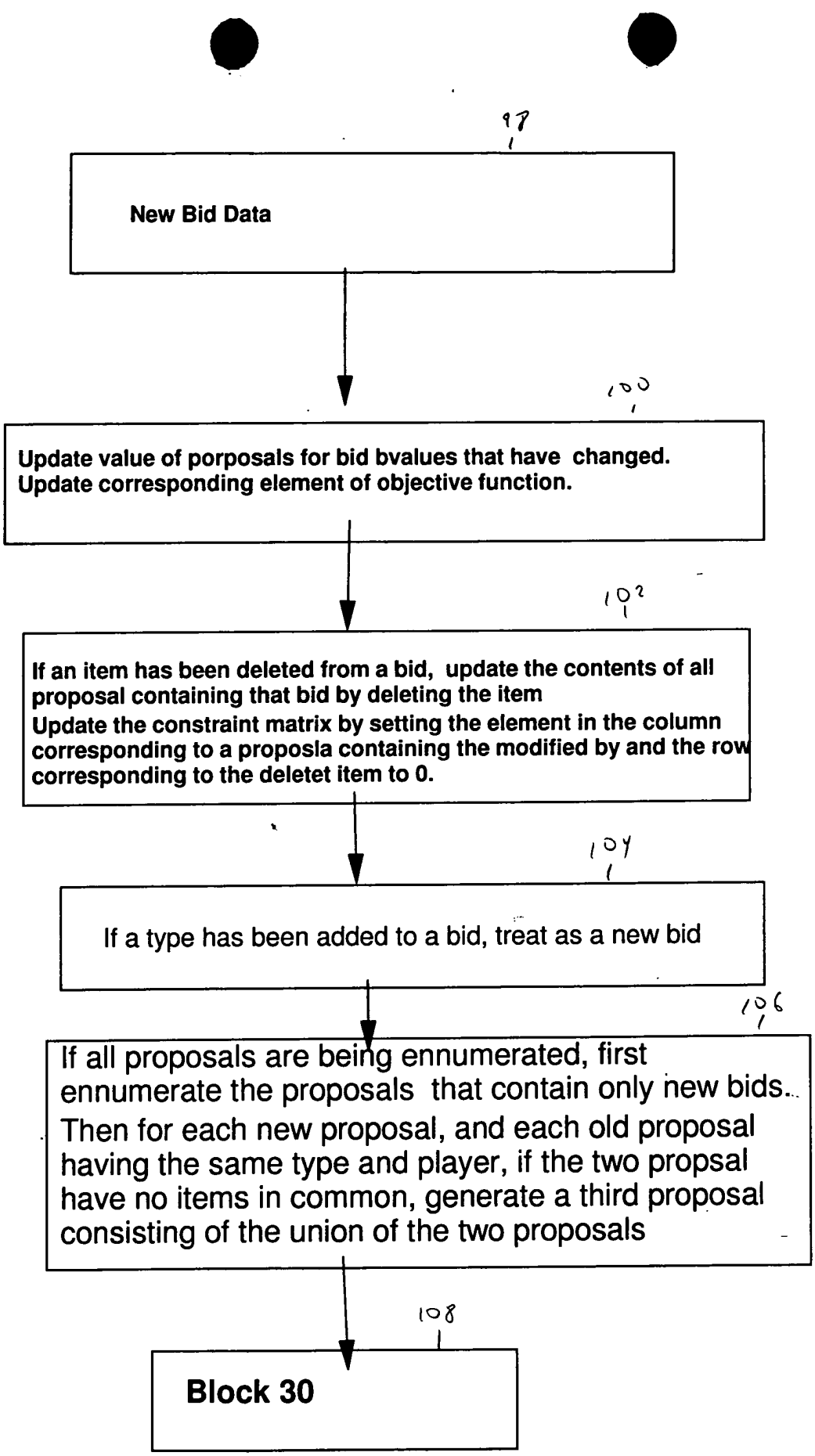
96
98
New Bid Data
100
Update value of porposals for bid bvalues that have changed.
Update corresponding element of objective function.
102
If an item has been deleted from a bid, update the contents of all proposal containing that bid by deleting the item
Update the constraint matrix by setting the element in the column corresponding to a proposla containing the modified by and the row corresponding to the deletet item to 0.
104
If a type has been added to a bid, treat as a new bid
106
If all proposals are being ennumerated, first ennumerate the proposals that contain only new bids.. Then for each new proposal, and each old proposal having the same type and player, if the two propsal have no items in common, generate a third proposal consisting of the union of the two proposals
108
Block 30

FIG 5